CLEAN SET OF CLAIMS

I claim:

1. A method of compressing data in a graphics processing system comprising:

    defining a plurality of tiles of data;

    defining a tile format table containing a status entry for each of said plurality of tiles;

    identifying the number of primitives contained in a tile;

    compressing said tile by storing said data in a plane equation format when said compressed tile is smaller than said tile;

    setting said status entry for said compressed tile in said tile format table; and

    storing said compressed tile in a memory.

2. The method of claim 1 wherein a tile is compressed when said tile has a number of primitives less than or equal to 1/3 of the number of pixels in said tile.

3. The method of claim 1 wherein said compression is lossless.

4. The method of claim 1 wherein each of said tiles comprises a cache line.

5. The method of claim 4 wherein pixels in said tiles represent Z data.

6. The method of claim 5 wherein said step of compressing comprises:

storing a plane equation for each primitive in said tile, wherein each primitive has a

fragment ID number; and

storing a fragment ID with each pixel in each said primitive.

7.   The method of claim 6 wherein said plane equation of pixels is computed by:

using the Z value of a first pixel as a first term of said plane equation;

subtracting X values of adjacent pixels in X that have the same fragment ID number to

obtain a second term; and

subtracting Y values of adjacent pixels in Y that have the same fragment ID number to

obtain a third term.

8.   The method of claim 2 wherein tiles read from said memory are decompressed

when said status bit indicates that said tile is a compressed tile.

9.   A graphics processing system comprising:

a first memory for storing a plurality of tiles of data;

a second memory for storing a tile format table containing a status entry for each of said

plurality of tiles; and

a compression system coupled to said first and second memories for compressing by

storing said data in a plane equation format said tile when said compressed tile is smaller than

said tile, for setting said status entry for said compressed tile in said tile format table, and for

storing said compressed tile in a memory.

10.   The graphics processing system of claim 9 wherein a tile is compressed when said tile has a number of primitives less than or equal to 1/3 of the number of pixels in said tile.

11.   The graphics processing system of claim 9 wherein said compression is lossless.

12.   The graphics processing system of claim 9 wherein each of said tiles comprises a cache line.

13.   The graphics processing system of claim 12 wherein said pixels in said tiles represent Z data.

14.   The graphics processing system of claim 13 wherein said compression system compresses by storing a plane equation for each primitive in said tile, wherein each primitive has a fragment ID number, and storing a fragment ID for each pixel in each said primitive.

15.   The graphics processing system of claim 14 wherein said plane equation of pixels is computed by:

using the Z value of a first pixel as a first term of said plane equation;

subtracting X values of adjacent pixels in X that have the same fragment ID number to obtain a second term; and

subtracting Y values of adjacent pixels in Y that have the same fragment ID number to obtain a third term.

3

16.   The graphics processing system of claim 11 wherein tiles read from said memory are decompressed when said status bit indicates that said tile is a compressed tile.

17.   A method of caching pixel data comprising:

storing pixel data in a plurality of tiles in a cache;

storing pixel data in a plurality of compressed tiles in a memory; and

decompressing a compressed tile of pixel data from said memory into said cache on a miss, wherein a fragment count number indicating the number of fragments in said compressed tile is recorded and a fragment ID number of each pixel in said tile is set to the fragment count number of the fragment which created said pixel.

18.   The method of claim 17 further comprises the step of:

writing Z data for a new primitive in a tile in said cache, wherein said fragment count number is incremented and pixels of said new Z data receive said incremented fragment count number as their fragment ID number.

19.   The method of claim 18 further comprises the step of:

evicting a tile by compressing said tile, said step of evicting further comprises:

determining if said tile is a good candidate for compression into a plane equation format;

using the Z value of a first pixel as a first term of said plane equation;

extracting values of pixels in X that have the same fragment ID number to obtain a second term; and

extracting values of pixels in Y that have the same fragment ID number to obtain a third

term.

20.   The method of claim 18 wherein said step of decompressing further comprises:

optionally decoding fragment ID mask that accompanies said compressed tile;

picking a first pixel of a first fragment ID and saving said coordinate of said first pixel as

a base tile coordinate;

using a plane equation to determine the Z values for all pixels with said first fragment ID;

and

repeating steps of picking and using for all fragments in said compressed tile.

21.   A pixel data cache comprising:

a cache storing pixel data in a plurality of tiles;

a memory component storing a plurality of compressed tiles whereby a compressed tile

of pixel data from said memory component is decompressed into said cache on a miss, wherein a

fragment count number indicating the number of fragments in said compressed tile is recorded

and a fragment ID number of each pixel in said tile is set to the fragment count number of the

fragment which created said pixel.

22.   The cache of claim 21 wherein said cache is configured to write Z data for a new

primitive in a tile in said cache, wherein said fragment count number is incremented and said

pixels of said new Z data receives said incremented fragment count number as their fragment ID number.

23.   The cache of claim 22 wherein said cache is configured to evict a tile by compressing said tile, wherein said tile is checked to determine if said tile is a good candidate for compression into a plane equation format and said cache is configured to use the Z value of a first pixel as a first term of said plane equation, extract values of pixels in X that have the same fragment ID number to obtain a second term; and extract values of pixels in Y that have the same fragment ID number to obtain a third term.

24.   The cache of claim 22 wherein said cache is configured to decompress by optionally decoding fragment ID mask that accompanies said compressed tile, picking a first pixel of a first fragment ID and saving said coordinate of said first pixel as a base tile coordinate, using a plane equation to determine the Z values for all pixels with said first fragment ID, and repeating steps of picking and using for all fragments in said compressed tile.